

# Curso de PHP

PHP

FATEC - Jundiaí

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Cookies

Cookies são arquivos texto que podem ser armazenados no computador do usuário, para serem recuperados posteriormente pelo servidor no qual seu site está hospedado.

Com os cookies podemos:

- Descobrir quantas vezes determinado usuário acessou seu site;
- Verificar se o usuário tem autorização para acessar uma página;
- Descobrir o nome de usuário dele e fazer o preenchimento automático do formulário de login (entrada em áreas restritas).

Podemos dividir a autenticação do usuário em três etapas:

- Criação de uma página de login, que recebe os dados do usuário, verifica se ele está cadastrado e cria os cookies.
- Criação de uma rotina de validação para ser utilizada nas páginas que fazem parte das áreas restritas.
- Criação de uma página de logout para os usuários que estão autenticados (realizar exclusão dos cookies).

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Cookies

Cookies são formas de armazenar informações a respeito de uma sessão dentro do disco rígido do usuário cliente. O comando `setcookie` armazena um cookie com as informações que se desejam recuperar em seguida. Quando não for declarado um tempo de vida, o cookie se auto-destrói quando a sessão é encerrada (quando o browser for fechado).

#### Sintaxe

```
setcookie(nome, valor, validade, caminho, nome_dominio, seguro)
```

**Nome:** indica o nome do cookie que está sendo enviado, e é o único parâmetro obrigatório.

**Valor:** deve ser enviado quando estamos incluindo um valor para um determinado cookie.

**Validade:** deve ser expresso no formato-padrão de tempo do Unix (número de segundos após o 1º de janeiro de 1970, às 0h).

**Caminho:** especifica o diretório (ou o nome inicial dos diretórios) em que o cookie é válido.

**Nome\_dominio:** indica o nome do domínio onde está hospedado seu site.

**Seguro:** é um valor inteiro (0 ou 1), que indica se o cookie é seguro. Se for utilizado o valor 1, o cookie só será transmitido se a conexão for segura (HTTPS).

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Cookies

Para incluir um cookie na máquina do usuário:  
`setcookie("nome", "PHP");`

Para excluir o cookie acima, basta aplicar este comando:  
`setcookie("nome");`

Para criar um cookie válido por 2 dias (48 horas), podemos utilizar como auxílio a função `time` do PHP.  
`setcookie("nome", "PHP", time() + 172800);`

Obs.: o envio de cookies deve ser a primeira coisa a ser feita na execução da página, ou seja, a função `setcookie` deve ser utilizada antes de qualquer tag HTML, como por exemplo, a tag `<HTML>`. Se for enviada após, o PHP gerará um erro. Devemos lembrar que os cookies não poderão ser utilizados dentro da própria página que os criou, somente a partir da próxima solicitação de página vinda do navegador do usuário.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Cookies

Existem duas maneiras de recuperar o valor de um cookie:

- 1) Usando o nome que foi dado dentro do comando setcookie, como uma variável global.

```
setcookie("nome", "PHP");
```

```
echo "O curso é de linguagem de programação em $nome<br>";
```

Obs.: o PHP define variáveis correspondentes a todos os cookies recebidos do navegador

- 2) Usando o array \$HTTP\_COOKIE\_VARS, utilizando o nome do cookie como chave associativa.

```
echo "O $HTTP_COOKIE_VARS["nome"] é uma ótima linguagem<br>";
```

Obs.: muitos usuários consideram os cookies como formas de invadir a sua privacidade. Portanto, se o seu site depender de cookies para funcionar, declare isso explicitamente.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Cookies

Múltiplas chamadas à função setcookie serão executadas em ordem inversa;

Cookies só podem ser gravados antes do envio de qualquer informação para o cliente. Portanto todas as chamadas à função setcookie devem ser feitas antes do envio de qualquer header ou texto.

Um problema com cookies é a segurança das informações, é extremamente fácil decifrar o conteúdo do cookie. Podemos obter o nome do usuário e senha, simplesmente abrindo o arquivo relacionado em um editor de textos qualquer.

```
userid"teste"www.tecspace.com.br/"0"84234894"u8d88992"teste"*"  
senha"1234"www.tecspace.com.br/"0"84234894"u8d88992"teste"*"
```

Os cookies no ambiente Windows são armazenados dentro do diretório cookies.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7


### Cookies

```
<?php
if (isset($_HTTP_POST_VARS['usuario'])) {
    $user = $_HTTP_POST_VARS['usuario'];
    setcookie("usuario", $user, time()+3600); // Expira em uma hora
    $mensagem = "Usuário $user conectado.<p>";
}
else {
    $mensagem = "Digite o seu nome de usuário<p>";
}
?>
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
<?php
    echo $mensagem;
?>
<form method="post" action="teste.php">
Nome de Usuário: <input type="text" name="usuario">
<br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

← Teste.php

Index.php

```
<html>
<head>
<title>Página PHP</title>
</head>
<body>
<?php
    $user = $_COOKIE["usuario"];
    echo "O usuário $user está conectado.";
?>
</body>
</html>
```



Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Sessões

Como o HTTP é um protocolo sem memória, cada invocação de cada script PHP é totalmente independente das outras invocações.

Isto significa que, em uma aplicação composta por vários scripts (o que é o normal), as variáveis não são preservadas de um script para o outro, e nem entre várias invocações do mesmo script.

Em vários casos (como no exemplo da agenda) a passagem de parâmetros via URL ou campos escondidos é suficiente para manter o contexto da aplicação.

Entretanto, em vários casos (como um carrinho de compras) não é possível, elegante ou seguro passar todo o contexto via URL ou campos escondidos.

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Sessões

`session_start()` cria uma sessão ou carrega o contexto de uma sessão previamente criada.

`session_destroy()` destrói uma sessão.

`session_id()` retorna o id da sessão corrente.

`session_register (var, ...)` registra uma ou mais variáveis globais como fazendo parte do contexto da sessão.

`session_unregister (var, ...)` retira uma ou mais variáveis globais do contexto da sessão.

`session_is_registered (var, ...)` verifica se as variáveis globais nomeadas fazem parte da sessão corrente.

Como por default a sessão utiliza cookies (que são cabeçalhos HTTP), devemos chamar `session_start()` antes de qualquer linha de texto fora do tag PHP

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Sessões

Programa login.htm

```
<html>
<head>
<title>Login do Sistema</title>
</head>
<body>
<form method="post" action="verifica_sessao.php">
  Usuário: <input type="text" name="login"><br><br>
  Senha: <input type="password" name="senha"><br><br>
  <input type="submit" value="Enviar">
</form>
</body>
</html>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Sessões

Programa verifica\_sessao.php

```
<?
if (($_POST["login"]=="claudio") && ($_POST["senha"]=="fatec")) {
    session_start();
    $usuario[] = $login;
    $usuario[] = $senha;
    session_register("usuario");
    echo "<a href='sistema.php'>Sistema</a>";
}
else {
    echo "<a href='login.htm' target='_parent'>Usuário Inexistente
ou senha invalida</a>";
}
?>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Sessões

Programa sistema.php

```
<?
session_start();
if (isset($_SESSION['usuario'])) {
    echo "Cadastro de Produtos<br>";
    echo "-----<br>";
    echo "Inclusão de Produto<br>";
    echo "Exclusão de Produto<br>";
    echo "Alteração de Produto<br>";
    echo "<a href='deslogar.php'>Sair do Sistema</a>";
}
else {
    echo "Usuário não está logado!";
}
?>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Sessões

Programa deslogar.php

```
<?
  session_start();
  if (isset($_SESSION['usuario'])) {
    session_unregister("usuario");
    session_destroy();
    echo "Usuário deslogado<br>";
  }
  echo "<a href='login.htm'>Login do Sistema</a>";
?>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Arrays Superglobais

Nome	Descrição
<code>\$GLOBALS</code>	Contém uma referência para todas as variáveis disponíveis dentro do escopo global do script.
<code>\$_SERVER</code>	Variáveis criadas pelo servidor web ou diretamente relacionadas ao ambiente de execução do script atual.
<code>\$_GET</code>	Variáveis postadas para o script via método HTTP GET.
<code>\$_POST</code>	Variáveis postadas para o script via método HTTP POST.
<code>\$_COOKIE</code>	Variáveis postadas para o script via cookies HTTP.
<code>\$_FILES</code>	Variáveis postadas para o script via transferência de arquivos HTTP.
<code>\$_ENV</code>	Variáveis disponíveis no script do ambiente de execução.
<code>\$_REQUEST</code>	Variáveis postadas para o script por todos os mecanismos de input, e que não podem ter seu conteúdo garantido de qualquer forma.
<code>\$_SESSION</code>	Variáveis que estão atualmente registradas na sessão do script.

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### \$GLOBALS

A superglobal \$GLOBALS, contém um referência para todas as variáveis que são atualmente disponíveis dentro do escopo global do script.

```
<?
$a = 5; // ESCOPO GLOBAL
$b = 3; // ESCOPO GLOBAL

function mostra_valor() {
    $GLOBALS["soma"] = $GLOBALS["a"] + $GLOBALS["b"];
}
mostra_valor();
echo $soma;
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### \$\_SERVER

São nada mais que, variáveis criadas pelo servidor ou diretamente relacionado ao ambiente de execução do script atual.

NOME	DESCRIÇÃO
PHP_SELF	Nome do arquivo do script atualmente em uso.
SERVER_NAME	Nome do servidor onde o script atual é executado.
SERVER_SOFTWARE	String de identificação do servidor.
SERVER_PORT	Porta usada pelo servidor para comunicação.
REQUEST_METHOD	Método utilizado para acessar a página. Exemplo: GET, HEAD, POST, PUT.
QUERY_STRING	String de solicitação pela qual a página foi solicitada.
DOCUMENT_ROOT	Diretório raiz onde o script atual é executado.
HTTP_REFERER	Endereço da página do qual o usuário acessou a página atual.
HTTP_USER_AGENT	Browser utilizado pelo usuário.
REMOTE_ADDR	Endereço IP do usuário.
REMOTE_PORT	Porta TCP utilizada pelo usuário para comunicação com o servidor.
SCRIPT_FILENAME	Caminho absoluto do script atual em execução.
SCRIPT_NAME	Caminho completo do script atual.

*Prof. Cláudio Farias Rossoni*



## PHP – Aula 7

### \$\_SERVER

```
<?
//DADOS DO USUÁRIO
echo "IP: $_SERVER[REMOTE_ADDR] <br>";
echo "NAVEGADOR: $_SERVER[HTTP_USER_AGENT] <br>";
echo "DIRETÓRIO RAIZ: $_SERVER[DOCUMENT_ROOT] <br>";
echo "CAMINHO DO SCRIPT: $_SERVER[SCRIPT_NAME] <br>";
echo "CAMINHO ABSOLUTO DO SCRIPT: $_SERVER[SCRIPT_FILENAME] <br>";
echo "IDENTIFICAÇÃO DO SERVIDOR: $_SERVER[SERVER_SOFTWARE] <br>";
echo "NOME DO SERVIDOR: $_SERVER[SERVER_NAME] <br>";
echo "PORTA SERVIDOR: $_SERVER[SERVER_PORT] <br>";
?>
```

#### Resultado no terminal seria:

```
IP: 127.0.0.1
NAVEGADOR: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
DIRETORIO RAIZ: c:/www
CAMINHO DO SCRIPT: /teste/teste.php
CAMINHO ABSOLUTO DO SCRIPT: c:\\www\\teste\\teste.php
IDENTIFICAÇÃO DO SERVIDOR: Apache 2.0.50 (Win32)
NOME DO SERVIDOR: tecspace
PORTA DO SERVIDOR: 80
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Variáveis enviadas pelo navegador - \$\_GET

O uso de parâmetros facilita a programação porque permite a passagem de dados entre o browser e o script ou entre scripts. A passagem de parâmetros entre o browser e o script é feita dentro da URL, por exemplo e é manipulada pela função \$\_GET.

Nesse exemplo a seguir, cada um dos links envia um valor diferente para a página que é aberta (teste.php). Para enviar um parâmetro, a sintaxe inclui um sinal de interrogação, o nome da variável, um sinal de igual e o valor da variável.

```
<html>
<head>
<title>Página PHP</title>
</head>
<body>
<?php
if (isset($_GET["valor"]))
{
$valor = $_GET["valor"];
echo "Você clicou no link $valor <p>";
}
else
{
echo "Clique em um dos links abaixo:<p>";
}
?>
<a href="teste.php?valor=1">link 1</a><br>
<a href="teste.php?valor=2">link 2</a><br>
<a href="teste.php?valor=3">link 3</a><br>
<a href="teste.php?valor=4">link 4</a><br>
<a href="teste.php?valor=5">link 5</a><br>
</body>
</html>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Variáveis enviadas pelo navegador - \$\_GET

Caso exista necessidade de se passar mais de um parâmetro, deve-se separá-los através de "e comercial" (&)

```
<html>
<head>
<title>Página PHP</title>
</head>
<body>
<?php
if (isset($_GET["produto"]) && isset($_GET["tipo"])) {
    $produto = $_GET["produto"];
    $tipo = $_GET["tipo"];
    echo "O produto selecionado foi $produto $tipo<p>";
}
else {
    echo "Selecione um produto<p>";
}
?>
<a href="teste.php?produto=Maxtor&tipo=HD">Maxtor HD</a><br>
<a href="teste.php?produto=LG&tipo=Monitor">LG Monitor</a><br>
</body>
</html>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Variáveis enviadas pelo navegador - \$\_POST

Os valores enviados através de um formulário podem ser recuperados pela variável pré-definida \$\_POST. Através dela é possível obter os dados que foram enviados através do método POST do HTML, bastando indicar o nome do campo do formulário.

No comando action do formulário, deve-se indicar a página PHP que irá receber os valores. O mesmo documento pode conter o código e o formulário.

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### Variáveis enviadas pelo navegador - \$\_POST

```
<html>
<head>
<title>Página PHP</title>
</head>
<body>
<?php
if (isset($_POST["pnome"]) && isset($_POST["snome"])) {
    $pnome = $_POST["pnome"];
    $snome = $_POST["snome"];
    echo "Olá $pnome $snome.<p>";
}
else {
    echo "Digite o seu nome.<p>";
}
?>
<form method="post" action="teste.php">
Primeiro Nome: <input type="text" name="pnome">
<br>
Sobrenome: <input type="text" name="snome">
<br><br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### File Uploads

Formulários HTML, quando utilizando o método POST, podem definir um <INPUT> do tipo "file", que exibe uma caixa de seleção de arquivos.

```
<html>
<body>
<h1>Enviar Arquivo</h1>
<hr>
<Form enctype="multipart/form-data" action="upload.php" method="post">
    Nome do arquivo: <input name="arquivo" type="file"><br><br>
    <input type="submit" value="Enviar">
</form>
</body>
</html>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 7

### File Uploads

O formulário é enviado junto com os dados do formulário, codificado como se fosse um arquivo anexado em e-mail.

O script PHP pode manipular o arquivo por intermédio do array `$HTTP_POST_VARS`

`$HTTP_POST_FILES["arquivo"]["name"]` fornece o nome do arquivo enviado

`$HTTP_POST_FILES["arquivo"]["size"]` fornece o tamanho do arquivo

`$HTTP_POST_FILES["arquivo"]["tmp_name"]` fornece o nome do arquivo temporário onde o PHP salvou o arquivo enviado

Note que "arquivo" é o nome (name) do tag `<INPUT type="file">`

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

### Arquivo upload.php

```
<html>
<body>
<h1>Dados do arquivo enviado:</h1>
<table border=0>
<tr> <td>Nome:</td>
<td><?= $HTTP_POST_FILES["arquivo"]["name"] ?></td></tr>
<tr> <td>Tamanho:</td>
<td><?= $HTTP_POST_FILES["arquivo"]["size"] ?></td></tr>
<tr> <td>Salvo como:</td>
<td><?= $HTTP_POST_FILES["arquivo"]["tmp_name"] ?></td></tr>
</table>
</body>
</html>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 7

PHP

Termino da aula

*Prof. Cláudio Farias Rossoni*